PATH PLANNING ALGORITHM FOR A CAR LIKE ROBOT BASED ON MILP METHOD

MOHD PAWZI BIN HJ MOHD SADRI

A project report submitted in partial
Fulfillment of the requirement for the award of the
Degree of Master of Electrical & Electronic Engineering

Faculty of Electrical & Electronic Engineering
Universiti Tun Hussien Onn Malaysia

JANUARI 2013

ABSTRACT

This project is presents an algorithm for path planning optimal routes mobile robot "like a car" to a target in unknown environment. The proposed algorithm allows a mobile robot to navigate through static obstacles and finding the path in order to reach the target without collision. This algorithm provides the robot the possibility to move from the initial position to the final position (target). The proposed path finding strategy is to use mathematical programming techniques to find the optimal path between to state for mobile robot designed in unknown environment with stationary obstacles. Formulation of the basic problems is to have the vehicle moved from the initial dynamic state to a state without colliding with each other, while at the same time avoiding other stationary obstacles. It is shown that this problem can be rewritten as a linear program with mixed integer / linear constraints that account for the collision avoidance. This approach is that the path optimization can be easily solved using the CPLEX optimization software with AMPL interface / MATLAB. The final phases are the design and build coalitions of linear programs and binary constraints to avoid collision with obstacles by Integer Mixed Linear Program (MILP). The findings of this research have shown that the MILP method can be used in the path planning problem in terms of finding a safe and shortest path. This has been combined with collision avoidance constraints to form a mixed integer linear program, which can be solved by a commercial software package.

ABSTRAK

Projek ini membentangkan algoritma bagi menghasilkan rancangan laluan optimum robot mudah alih "seperti kereta" kepada sasaran dalam persekitaran yang tidak diketahui. Algoritma yang dicadangkan itu membolehkan robot mudah alih untuk bergerak melalui halangan pegun dan mencari jalan untuk mencapai sasaran tanpa perlanggaran. Algoritma ini menyediakan robot kemungkinan untuk bergerak dari kedudukan awal ke kedudukan akhir (sasaran). Strategi mencari jalan yang dicadangkan adalah dengan menggunakan teknik pengaturcaraan matematik Mixed Linear Integer Program (MILP) untuk mencari jalan yang optimum. Objektif utama projek ini adalah mencari laluan robot tanpa pemandu dari satu destinasi ke destinasi yang lain tanpa berlanggar antara satu sama lain, manakala pada masa yang sama mengelakkan halangan pegun lain. Ia menunjukkan bahawa masalah ini boleh ditulis semula sebagai program linear dengan kekangan integer / linear bercampur bagi mengelakkan perlanggaran. Pendekatan ini dapat mengoptimuman laluan dengan mudah. Penggunakan perisian CPLEX pengoptimuman dengan AMPL antaramuka / MATLAB digunakan bagi menyelesaikan semua persamaan mateatik . Fasa terakhir adalah reka bentuk dan membina pakatan program linear dan kekangan binari untuk mengelakkan perlanggaran dengan halangan oleh Program Integer Campuran Linear (MILP). Dapatan kajian ini telah menunjukkan bahawa kaedah MILP boleh digunakan dalam masalah perancangan jalan dari segi mencari jalan selamat dan terpendek. Ini telah digabungkan dengan kekangan mengelakkan perlanggaran untuk membentuk program bercampur integer linear, yang boleh diselesaikan oleh pakej perisian komersil.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Project Background

In recent years, demands for an autonomous robot carrying out activities that are limited to humans' ability are great. Designing such robot requires meticulous calculation and algorithm. Path planning algorithm is an example that can be implemented in such design [1-3].

Path planning algorithm for a robot is to generate a collision free path in an environment with obstacles and optimize it with respect to some criterion [6,7]. This algorithm requires ones knowledge on how to model a multi-obstacle avoidance environment in terms of mathematical programming techniques [4]. Mixed Integer Linear Programming (MILP) is an example of a mathematical programming techniques used by many [3,5]. The programming helps in finding an optimal path between two states either for a vehicle or a group of vehicles.

The path resulted from MILP is fuel-optimal, having the vehicles move from an initial dynamic state to a final state without colliding with each other, while at the same time avoiding other stationary and moving obstacles [1]. Even though MILP shows significant results in finding the optimal path, but the computational is too complex and time consuming [3]. Previous efforts have studied on path planning algorithm such as for climber robot [2], wheeled robot [3], unmanned air vehicles [5] and multiple vehicles [1,4] – be it in a 2D or 3D environment, but few had studied in developing a technique that requires less binary variables for generating a collision free trajectories. Trajectory planning is to schedule the movement of a mobile robot along the planned path. .

Several approaches have been proposed to address the problem of motion planning of a mobile robot. If the environment is a known static terrain and it generates a path in advance it said to be off-line algorithm. It is said to be on-line if it is capable of producing a new path in response to environmental changes. The focus of this project concerns on stationary obstacle in off-line.

1.2 Problem Statements

The main problem in manoeuvring an autonomous robot is how to ensure that the vehicles will not collide with each other or with obstacles; be it as a moving or stationary and the path trajectory is the optimal path. Previous methods include probability road map, artificial potential field and inspired biological intelligence algorithm. Although the former method is provably probabilistically complete, but it does not deal with the environment where the information is time-varying. For example, if an uncertainty appears in the robot workspace, probability road map cannot update with the changing environment and plan a valid trajectory for the mobile robot [5]. On the other hand, the artificial potential field works effectively in a dynamic environment but it has the property drawback of failing in determining the desired optimal trajectories. Whereas the inspired biological intelligence algorithm accommodates these two factors, but the computing takes forever even infinite time to find the best solution. Furthermore, all of them have a great number of parameters to tune and that is never an easy job particularly when the users are short of prior knowledge.

Thus in this project, an alternative approach is introduced in achieving those two intentions of manoeuvring an autonomous robot based on [8]. The optimal path planning is found through mathematical programming techniques (linear technique) while the collision avoidance is formulated through integer programming. This approach used the combination of these two programming techniques hence the name Mixed Integer Linear Programming (MILP). The study of designing the most optimal path planning algorithm is very important as its application can spread into many field such as oil drilling, wall-climbing robots for cleaning and painting city tall buildings, cooperative reconnaissance, air traffic management, coordinated robot motion, unmanned aerial vehicle (UAV) path planning and cooperative control [4]. And likely, the results of this study can contribute a significant insight and helps those who involved in designing and engineering such applications.

1.3 Project Objectives

The objectives of this project are:

a) To implement an MILP algorithm for optimal path planning
b) To apply the algorithm for obstacles collision avoidance
c) To analyze the performance of the designed MILP path planning algorithm on a car like robot

1.4 Project Scopes / Constrains

To ensure that the objectives of the project are achieved, a few important aspects are considered. The scopes for this project are:

a) The optimal path planning between two states are for a single vehicle
b) The algorithm can distinguish collision avoidance with stationary obstacles
c) The testing performance of the car liked robot is performed in a 2D environment
d) The environment in which path planning takes place is static.
e) The obstacles shape and location are known.

.

1.5 Thesis Outlines

This project thesis is done to basically document the concept, activities and outcome of the project that is relevant to the progress. The thesis stresses more on the activities that have been done in order to complete the project. This project thesis consists of main chapters.

a) Chapter one briefly describes about the project's introduction. It also discuss on the objectives and the scopes of the project.
b) Chapter two describes the explanation is stress more on the literature review that consists of an overview of Search Space Representation and an overview of Search Algorithm.
c) Chapter three explains the method used in implementing the project task. The technique and methodology of this project.
d) Chapter four discusses about the simulation result. In this chapter, each part will be described in depth.
e) Chapter five discusses about analysis and discussion of the project. In this chapter, all of final result and analysis that have been done will be stated clearly.
f) Chapter six is for the conclusion and recommendations. A conclusion about the achievement of the project is stated in this chapter. The recommendations are made to improve the project operation for the future work.

1.5 Summary

This chapter of this thesis discusses about the introduction for the whole project. Firstly, the concept of the path planning by MILP method introduced in the first part. Next, the problem statement is discussed. Then, the next part is about the objectives and scopes of the project. Lastly, the thesis outline is discussed which will give an overview for the reader about the thesis.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter summarizes the relevant information and literature path planning method for mobile robot in a static environment. This chapter also describes how the entire project was carried out, including the knowledge and skills acquired to complete the project. The main reference for this project and thesis projects related earlier. Other sources of literary journals, articles, and information from books and the internet. Those resources will help to identify the problems and come up with ideas for analysis and decision making in the project.

A basic mobile robot path planning task is to find a collision-free path through an environment with obstacles, from a certain point began over the desired point while satisfying certain optimization criteria. This chapter classifies the various methods of robot path planning in a different way and gives some general information about the traditional path planning methods in the same environment. There are two stages to find a route from the starting point to the ending point. The first stage is to represent the work area while the second is to choose the right path using graph search algorithms.

2.2 Configuration Space (C- Space)

As a robot is normally not a point in space, it cannot travel to some points in free space because of it's geometry. In order to solve this issue, the geometry of obstacles is transformed into new shapes according to robot's geometry so that robot can travel all around in new free space without any collide. New obstacle space is called configuration space (*C*-space) obstacles.

There are a number of environment representations based on *C*-space. The most common representations are explained in the following sections, based on the scenario represented by the *C*-space as depicted in Fig. 2.1 (b).

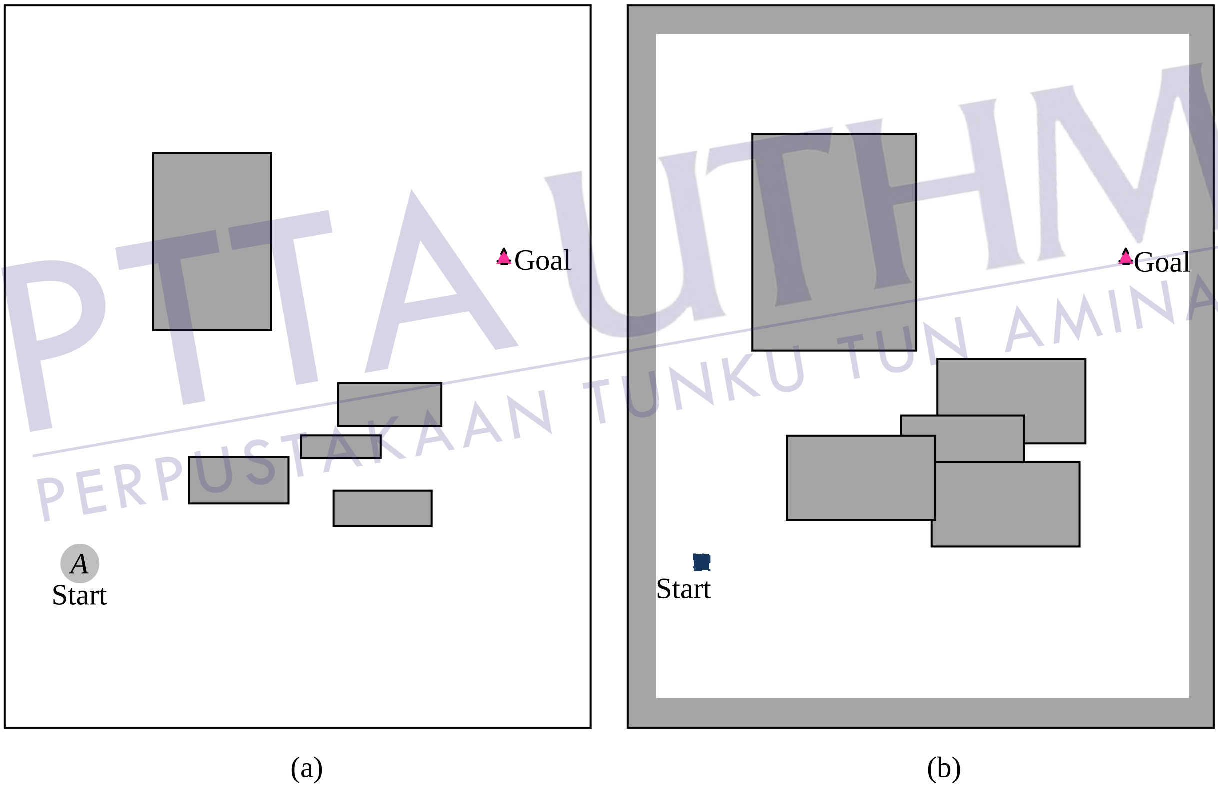


Figure 2.1: A scenario represented in (a) original form (b) configuration space. The darker rectangles in (a) are those with actual dimensions while in (b) are those enlarged according to the size of vehicle *A*. The white areas denote the free space.

2.3 Linear Programming / Mixed Integer Linear Program (MILP)

This is a method to implement path planning for a mobile robot by using mathematic programming. Based on the relative velocity coordinates (RVCs), linear programming (LP) and mixed integer linear programming (MILP) can be employed for path planning problem. For example, Wang et al. [7] converted the 2D path planning of an unmanned underwater vehicle to constrained optimization or semi-infinite constrained optimization problem. Zu et al. [8] discussed the path planning in 2D and an LP method was proposed for the problem of dynamic target pursuit and obstacle avoidance. This method tried to plan the variables of the linear acceleration and the angular acceleration of a ground vehicle.

The approach presented in this project formulates the problem as a mixed integer linear program (MILP). This is a modification to a linear program (LP) in which some variables are constrained to take only integer values. In particular, we use binary variables, taking only the values 0 or 1. Constraints on such variables enable the inclusion of discrete decisions in the optimization [9, 10], encoding the non-convexity of the problem. Both avoidance and assignment constraints can be considered in terms of such decisions. For collision avoidance, a vehicle must either be "left" or "right" of an obstacle, each leading to a convex sub-problem. Assignment can be expressed as discrete choices of destinations. Constraints on the binary variables are used to include logical requirements on the decisions, such as compatible assignments.

In general, MILPs are also NP-complete [11], indicating that the MILP representation retains the inherent complexity of the problem. However, in many instances, MILPs can be solved using a branch and bound algorithm, exploiting their relaxation to LP form to accelerate the solution process. The MILP form of the trajectory optimization problems is linear by definition, so the method is immune to issues of local minima and globally-optimal solutions can be found. Highly optimized, commercial software is available for this process. These codes were developed to solve MILPs in the field of operations research, such as airline scheduling [12]. The CPLEX optimization software [13] is used to solve the MILPs in this project, although various other options exist. CPLEX implements the branch-and-bound algorithm in conjunction with many adjustable heuristics, allowing quite large problems to be solved in practical computation times. There are two major drawbacks to the MILP approach. The first is the intensive nature of the computation, which is centralized and scales poorly with problem size. The second is the restriction to linear problems.

2.4 The Branch and Bound Search

Integer and mixed-integer programs are commonly solved by a solution method known As the Branch and Bound method. According to Winston ~~(1991)~~[xxx] the first step of the Branch and Bound method involves solving the LP relaxation. A common LP relaxation is the problem obtained by treating the integer variables as continuous. By definition of the relaxed problem, the feasible region of any mixed-integer linear program is contained within the feasible region of the LP relaxation. Additionally, this implies that the optimal function value of the LP relaxation will be greater than or equal to the optimal value of the mixed-integer linear program function (if dealing with a maximization objective). Hence, the solution of the LP relaxation can be regarded as a maximum or upper bound of the mixed integer linear program. In addition, this LP relaxation forms sub-problem o of Figure 2.2. If the optimal solution of the LP relaxation has all decision variables as an integer (or binary) value, then this is also the optimal solution of the mixed-integer linear program. Assume however, that this upper bound does not meet this criterion.

In order to find the optimal mixed-integer linear program solution, the feasible region of the LP relaxation must be partitioned. A relaxed integer variable that takes on a fractional value in the LP relaxation optimal solution (subproblem 0) is selected as a branching point. This variable is branched to the upper and lower integer value associated with the current fractional value; these integer values are included as upper and lower bounds respectively in the corresponding node subproblems.

In Figure 2.2, X2 has been selected as the branching point, which then creates subproblems 1 and 2. Arbitrarily, one of subproblems 1 or 2 is selected as the next branching point. Clearly, the result of each branch is two new subproblems. At each new pair of subproblems, this procedure is repeated, selecting one variable to further branch based on a fractional decision variable.

This process continues until one of three situations occurs:

- A candidate solution is reached. A candidate solution is one that has all (required) decision variables assigned integer values.
- The optimal function value of the subproblem does not exceed that of the current lower bound. A lower bound is the candidate solution with the highest objective function value that still remains less than or equal to that of the optimal LP relaxation solution (in the case of a maximization problem).
- The subproblem is infeasible (i.e. the constraints cannot be simultaneously satisfied) . Upon any of these three events occurring, no further branching is required from that node. Once the search-tree is complete, the optimal solution can easily be selected from the few candidate solutions (based on objective function value and feasibility)

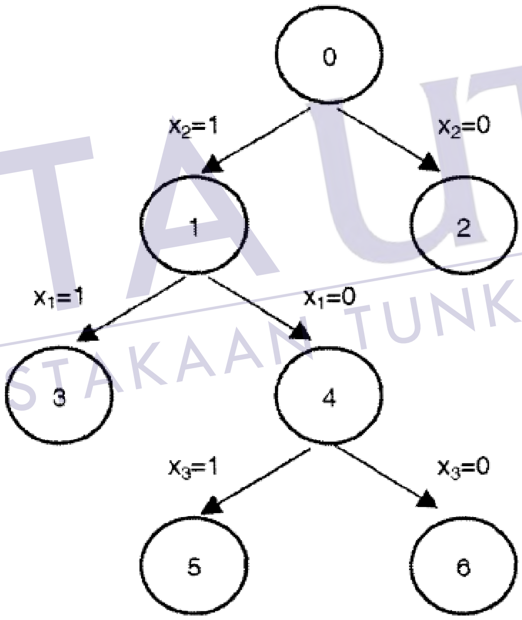


Figure 2.2: A binary search tree for the branch and bound method

2.5 Other Method Search Space Representation

In term of routes finding, a search space is represented by different approaches as shown in Fig. 2.3. This step will be applied to find a collision-free route, through an environment with obstacles, from a specified starting point to a desired ending point while satisfying certain optimization criteria. Some are applicable to a wide variety of path planning problems, where others have a limited applicability [2]. Most of path planning problem can be solve using these approaches. This algorithm can be use depending on the situation and purpose. Search space Representation has many approaches but in this thesis describes only four approaches.

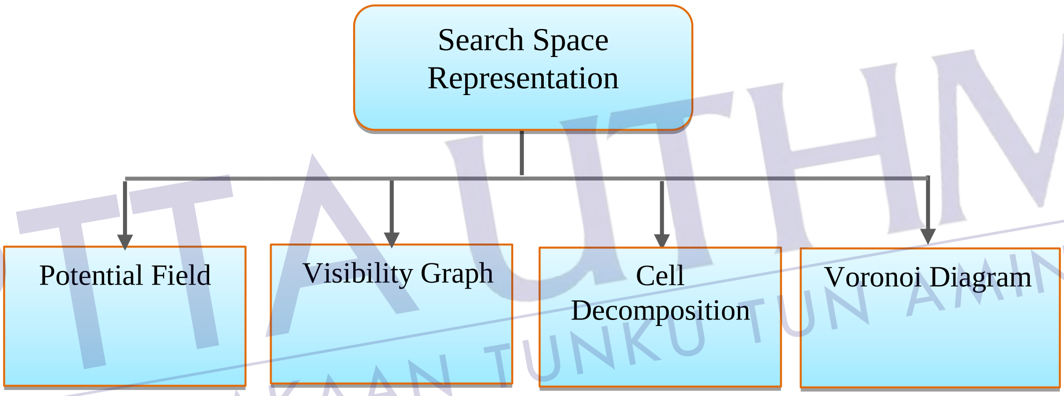
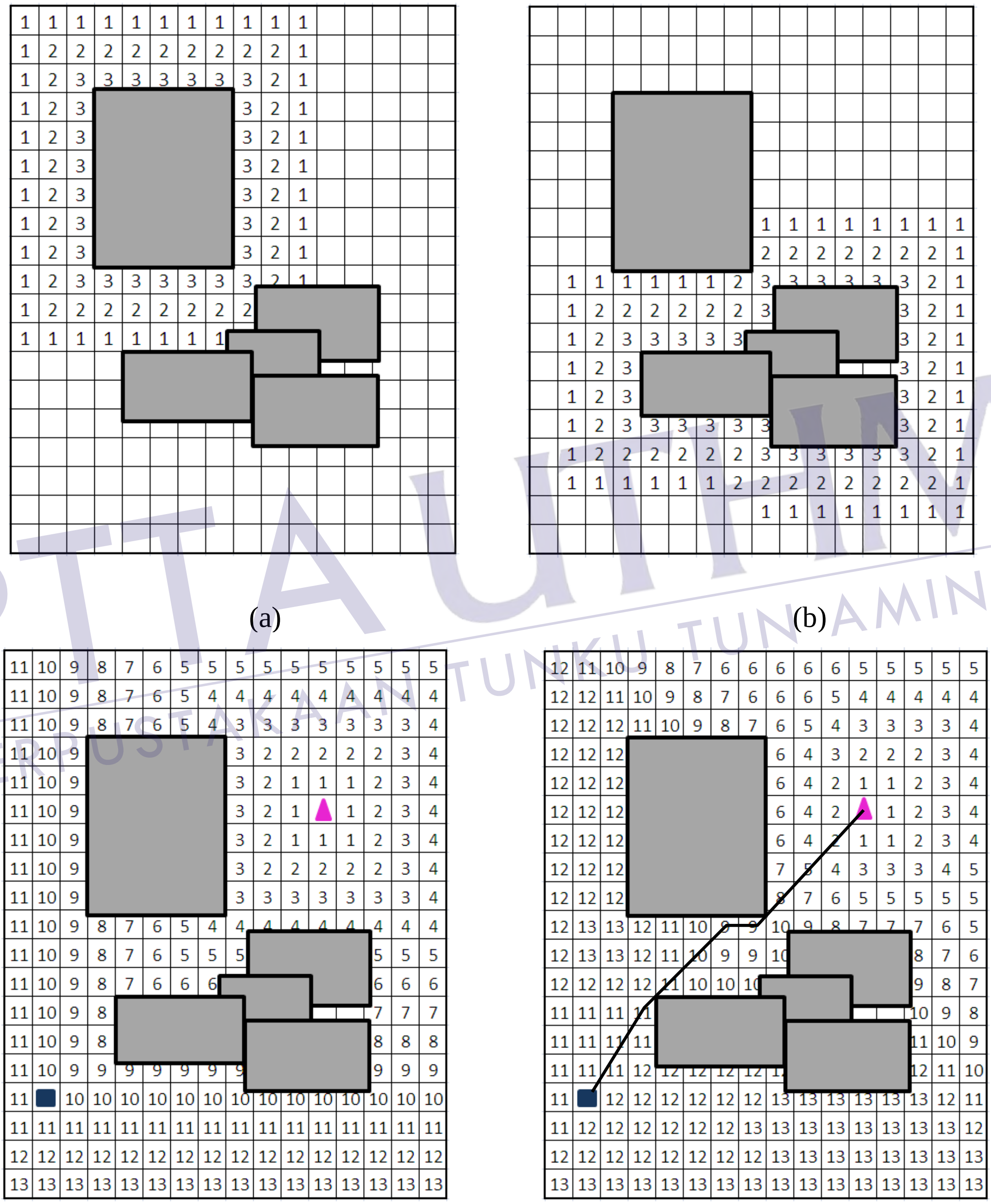


Figure 2.3: Classifications of the Search Space Representation methods

2.5.1 Potential Field (PF)

Potential field (PF) method has been extensively used in path planning by many researchers. This idea was first suggested by Khatib [14]. PF treats a vehicle as a point under the influence of fields which are generated by the starting point (p_{start}), target point (p_{target})and the obstacles in the workspace $\mathcal{W}$. In other words, the vehicle acts like an electron in an electric field with repulsive and attractive forces. Repulsive forces are generated by obstacles and p_{start} while attractive forces are generated by p_{target}. The resultant force of the field on the vehicle determines the vehicle motion direction. There are three steps in potential field method to accomplish a path planning process. The first one is to generate a force to guide the vehicle towards the goal. The second step is to design forces to guide the vehicle away from obstacles and the third step is to apply the total force to the vehicle. An example of discrete potential field representation used in path planning is shown in Fig. 2.4.



(a) (b)

(c) (d)

Figure 2.4: Simplified potential fields. Field produced by obstacles in (a) and (b), the field produced to create goal attraction in (c), and the sum of the fields in (d). This summed field will be used to direct the vehicle along the levels of lowest potential.

One major disadvantage of potential field method is the problem of containing local minima other than the goal at which the vehicle can get trapped. As potential field method directs the vehicle towards a minimum in the field, it is not guaranteed that the minimum is the global minimum. This makes the potential field a local method, as opposed to a global method, because the effect of the field on the vehicle is almost exclusively based on obstacles near it. Obstacles far away have small or even no effect on the vehicle's motion. Another drawback of the potential field method is unsatisfactory obstacle avoidance when the vehicle maneuvers through tight environment [15]. This is due to the potential fields are typically modeled as continuous and differentiable functions, leading to an imprecise description of the obstacle's shape and dimension. Readers are referred to [16] for a comprehensive continuous potential field description.

2.5.2 Related Works using Potential Field

Khatib [14] reported the earlier result of real-time obstacle avoidance for mobile robots based on potential field (PF) method. In this method, a mobile robot moves in a PF such that the goal is modeled as an attractive force and the obstacles and other vehicles are modeled as repelling forces. This method is then expanded by many others. Yong *et al.* [17] discussed a path-planning algorithm in 3D environment using a potential field representation of obstacles. The proposed path planning algorithm consists of two levels, the first being a robot's path and its orientations along the path are selected by a global planner from the minimum potential valleys. Then the path and orientations are modified by local planner to derive the final collision-free path and orientations. However, the algorithm fails to plan in tight free spaces. It also cannot be efficiently applied for real-time path planning. A work by Poty *et al.* [18] considered real-time path planning problem of mobile robots in a dynamic environment with dynamic obstacles based on potential field method. The *Ge* and *Cui* method is introduced to represent the attractive and repulsive potentials considering the dynamic position and velocity of the robot and obstacles. Using computer simulations, it is proven that the robot successfully avoids obstacles and reaches at the target. Another work on real-time path-planning is by Chen et. al [19] who proposed a method based on behaviour information potential field (BIPF) in unknown environments. The idea is based on building an imaginary field to represent the robot's behaviour. They used a 2-D Cartesian Histogram Grid as an imaginary field model which is updated continuously based on the robots actions. The robot has the tendency to observe the region with less BIPF value. [19] claims that BIPF has higher efficiency and better performance than the traditional path planning methods such as PRM.

Many efforts have been undertaken to address the local minima problem associated with PF method. Khosla and Volpe [20] proposed a PF method based on super-quadratic potential function, which is used to model obstacles with arbitrary shapes. This approach removes the local minima but with less success and increases computational effort. On the other hand Kim and Khosla [21] introduced the harmonic functions to build a potential field in the workspace without local minima. Akishita *et al*. [22] proposed potential field method that incorporates Laplace equation for path planning of a mobile robot. Laplace function's solution was utilised in finding a smooth and collision free path. The potential field is generated over the entire C-space and the harmonic solutions to Laplace equation are utilized to find the path for the robot from a starting to target points. Daily and Bevly [23] extended the work of [21] for high speed vehicles path planning. Another work on solving the local minima issue of potential field method is done by Wang and Chirikjian [7] who proposed a modified artificial potential field method for path planning of non-spherical single-body robots, motivated by steady-state heat transfer with variable thermal conductivity technique. In order to describe the obstacles and free-space, the variable thermal conductivity was applied. The optimal path problem is the same as a heat flow with minimal thermal resistance. The advantages using the proposed approach are freedom from local minima, smoothness, optimality and collision-avoidable. This approach can be applied to path planning in both static and dynamic environments.

The traditional method, such as probability road map, can achieve a successful path in 2D static environments. The planning process using this method generally consists of two phases : a construction and a query phase. In construction stage, the workspace of the robot is sampled randomly for generating candidate waypoints. In the query stage, the waypoints between the start and goal position are connected to be a graph, and the path is obtained by some searching algorithm, such as Dijkstra, A* algorithm and so on.

Hraba researched the 3D application of probability road map where A* algorithm is used to find the near-optimal path (Hrabar, 2006). Although probability road map method is provably probabilistically complete (Ladd & Kavraki, 2004), it does not deal with the environment where the information is time-varying. The underlying reason is that this method only focuses on the certain environment. Once some uncertainty appears in the robot workspace, probability road map cannot update with the changing environment and plan a valid trajectory for the mobile robot, never an optimal path.

Artificial potential field is another traditional method which is generally used in both 2D and 3D environment. The mechanism that the robot is driven by attractive and repulsive force in a cooperative way is simple and often works efficiently even in dynamic environment. Kitamura et al. construct the path planning model based on the artificial potential field in three- dimensional space which is described by octree (Kitamura et al, 1995). Traditionally, artificial potential field applies in two dimensions extensively. Also some other field concepts are invented. For example, there are harmonic potential functions (Kim & Khosla, 1992; Fahimi et al, 2009; Cocaud et al, 2008; Zhang & Valavanis, 1997), hydrodynamics (Liu gradient field (Konolige, 2000), and virtual force field (Oh et al., 2007).

Unfortunately, path planning approach based on the function family of potential field cannot obtain the optimal objective function which, in turn, cannot guarantee the desired optimal trajectories. Additionally, some of them, for example, potential panel method and harmonic potential function, can plan a path for an autonomous vehicle, but the computing burdens is huge and real time performance hardly satisfies the practical requirement.

Inspired by biological intelligence, many approaches, such as ant colony optimization (ACO) (Chen et al., 2008), particle swarm optimization (PSO) (Jung et al., 2006), genetic algorithm (GA) (Allaire et al., 2009), evolution algorithm (EA) (Zhao & Murthy, 2007), and their combination, are introduced to solve the path planning problem. They mostly rely on the stochastic searching, known as non-deterministic algorithm. These methods will eventually find an optimal solution, but no estimate on the time of convergence can be given. Thus, it may take long even infinite time to find the best solution. Furthermore, all of them have a great number of parameters to tune and that is never an easy job particularly when users are short of prior knowledge. Some comparisons (Allaire et al., 2009; Krenzke, 2006) show that the expensive calculations limit their real application.

2.5.3 Visibility Graph (VG)

In computational geometry and robot motion planning, a VG is the collection of lines in the free space that connect a feature of an object to that of another object [6] as shown in Fig. 2.5. VG is very useful variations of sampling-based roadmaps but works very hard to ensure that roadmap representation is small [7]. The roadmap, each node in the graph represents a point location, and each edge represents a visible connection between them. That is, if the line segment connecting two locations does not pass through any obstacle, an edge is draw between them in the graph [8]. The advantages of using VG are it is simple [2] and a well-known path planning approach [9]. This yields optimal paths in 2D or 3D [10]. The defect of the VG approach is that the efficiency of the algorithm is low. The robot may lead to crashing if the obtained path is often very close to obstacles [11]. In term of calculation, this approach is poor because the calculated paths are tangential to the obstacles and the robot will brush right up against the obstacles. The obstacles must be clearly defined polygon so for the outdoor robot, there have a problem since the obstacles usually take on round [8]. This approach do not perform well in dynamic environments and difficult to gain the optimal path for the robot. Other than that, in another research was founded the visibility line need a lot of time to processing when the number of obstacles in the search space is increase [12].

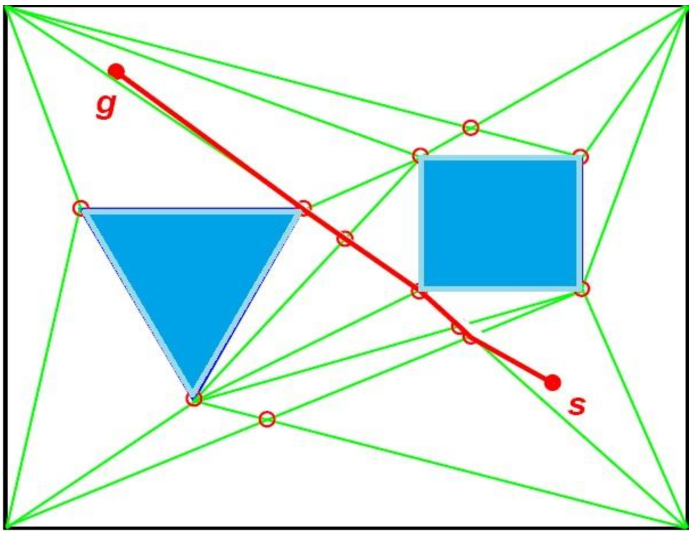


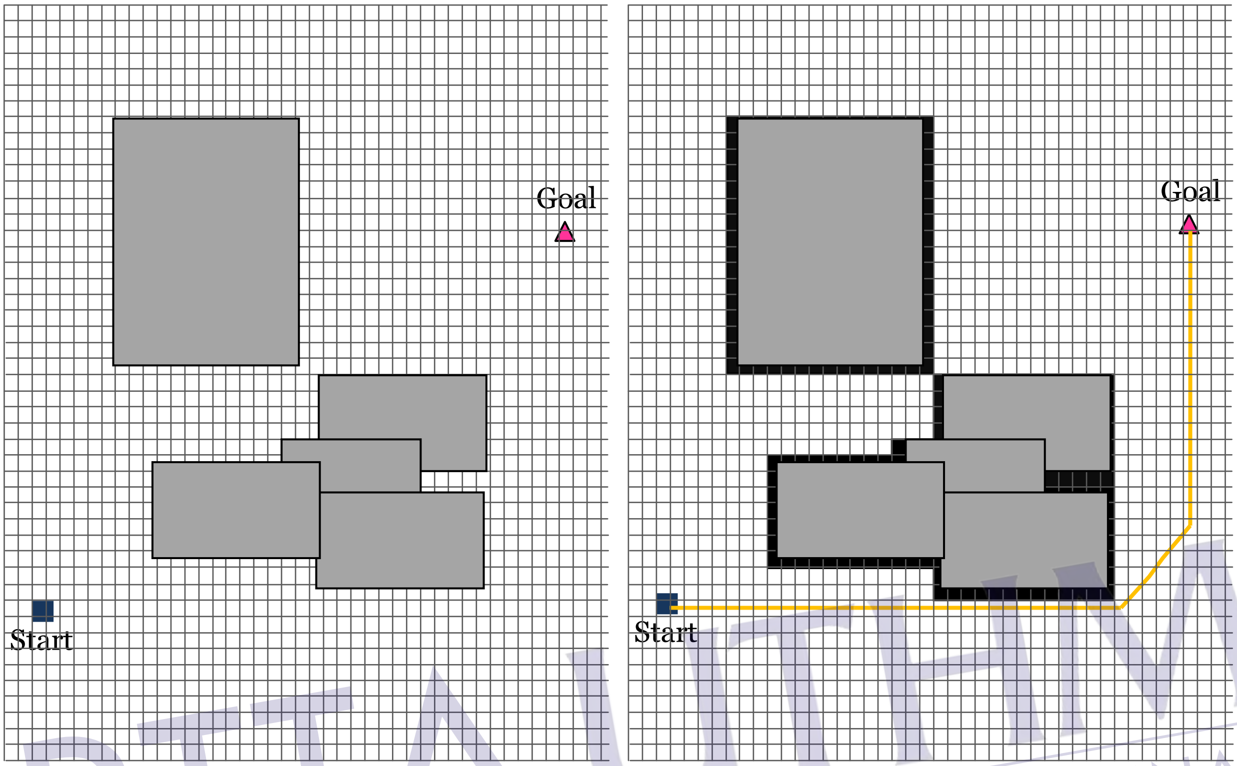
Figure 2.5: The visibility graph

2.5.4 Cell Decomposition

Cell decomposition (CD)-based are one of the popular representation methods especially in outdoor environment [12]. In CD, C-space is first divided into simple, connected region called cells [24]. The cells, which are discrete, non-overlapping but adjacent to each other, might be in rectangular or polygonal shape. If the cell contains obstacle (or part of obstacle), it is marked as occupied, otherwise it is obstacle free. A connectivity graph is then constructed to link the adjacent cells. There are several variants of CD including Regular Grid, Adaptive Cell Decomposition and Exact Cell Decomposition.

Regular Grid

Regular grid (RG) technique was introduced by Brooks and Lozano-Perez [13] to find a collision-free path for an object moving through cluttered obstacles. In general RG can be constructed by laying a regular grid over the configuration space. RG is easy to apply because of the shape and size of the cells is predefined. RG samples the world and marks up the graph accordingly as to whether the space is full, empty or partially full. A cell is marked as an obstacle if an object or part of it occupies the cell. Otherwise it is regarded as Q_{free}. The centre of each free space cell represents a node in the C-space. Connectivity graph is then constructed from all the nodes. This method is popular due to its simplicity in application to a C-space. The computation time can be reduced by increasing the cell size, thus reducing the number of cells to search through. On the other hand the cell size can be made smaller in order to provide more completeness and detail. Example of a path planning using RG is illustrated in Fig. 2.6(a). The path connecting the starting point and target point is shown in solid yellow line. Although RG is easy to apply, there are some drawbacks with this method. First, the digitisation bias where an obstacle that is much smaller than the cell size will result in that entire grid square being labeled as occupied [12]. Consequently, a passable space might be considered impassable by the planner as illustrated in Fig. 2.6(b). Furthermore, if the cell is too big (hence grid resolution is too coarse), the planner might not be complete i.e. finding a path where one exists is not guaranteed. Another drawback of RG is its inefficiency in representing the C-space as in sparse area many equally sized cells are required to fill the empty space. As a result, planning is expensive because more cells are processed than actually required [49].



(a) (b)

Figure 2.6: (a) The *C*-space obstacles. (b) The obstacles represented by regular grid techniques. In (b) the passable area is considered impassable.

Adaptive cell Decomposition

Unlike RG, adaptive cell decomposition (ACD) is built using quad-tree. The cells of a quad-tree are identified either as free cells, which contain no obstacles; or as obstacles cells, where the cells are occupied; or as mixed cells, which represent nodes with both free space and obstacles. The mixed cells have to be recursively sub-divided into four identical sub-cells until the resulted smaller cells contain no obstacles' region or the smallest cells are produced [19, 25]. The result of ACD is a map that has grid cells of different size and concentration with the cell boundaries matched the obstacle boundaries closely. It produces lesser number of cells so that the C-space can be used more efficiently. This requires less memory and processing time. It also removes the digitisation bias of RG. An example of ACD representation used for path planning is depicted in Fig. 2.7(a) The collision-free path that connects starting point (Start) and target point (Goal) is depicted in solid yellow line.

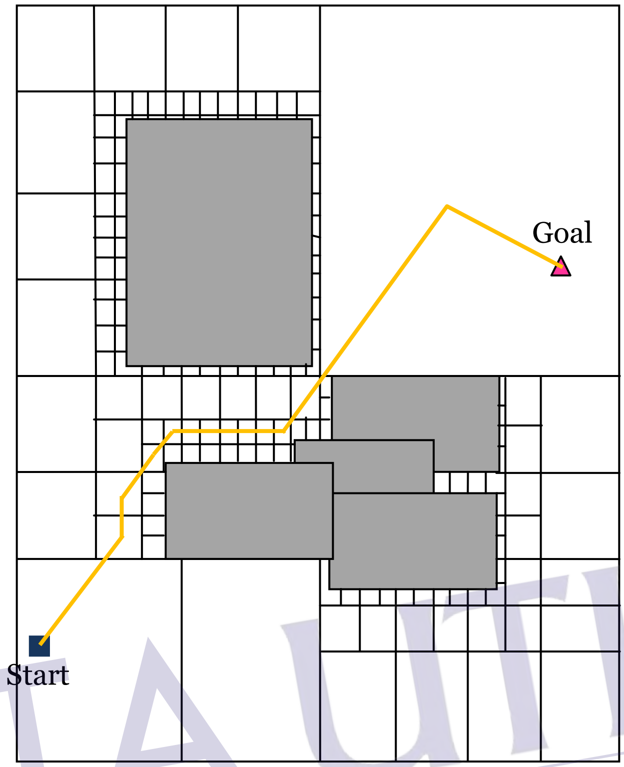


Figure 2.7(a) : Path planning using framed quad-tree

However, ACD would not efficient in dynamic environments where the vehicle is obtaining new data and updating its map based on new obstacles. In order to address this problem, the whole data structure of the map must be completely revised [12]. In addition, quad-trees often result in jagged.

Chen et.al [48] has proposed a good solution to this problem called framed quad-trees as shown in Fig. 2.6(b). Unlike standard quad-tree, framed-quad-tree has a border array of the smallest possible allowed size cells, which is called frame. The path generated by framed quad-trees is shown in solid yellow line. Comparing both the paths as in Figs. 2.6(a) and 2.7(b), it is clear that the quality of the framed quad-tree path is higher than that of ACD. Nevertheless, quad-trees can be less efficient than regular grids in high clutter environments due to the overhead required to keep track of the cell sizes and locations.

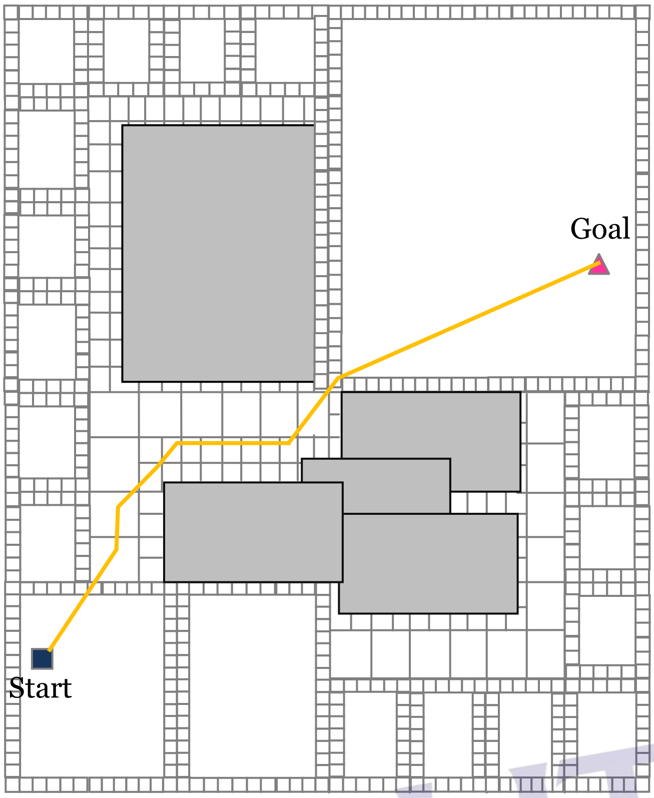


Figure 2.7 (b) : Path planning using framed quad-tree

Exact Cell Decomposition

Another variant of CD is Exact Cell Decomposition (ECD) method which consists of two-dimensional cells, used to solve some of the problems associated with regular grids. The sizes of the cells are not predefined, but are decided based on the location and shape of obstacles within the C-space [12]. The cell boundaries are the boundaries of the C-space, and the union of the cells is exactly that of the Q_{free}. Therefore, ECD is complete i.e. always find a path if one exists. Example of ECD is shown in Fig. 2.8. The path connecting starting (Start) and target (Goal) points is shown in solid yellow line.

However, the paths generated by ECD are not optimal (shortest) and there is no simple rule on how to decompose the space into cells. This method is not suitable for outdoor environments where obstacles are often poorly defined and of irregular shape [12].

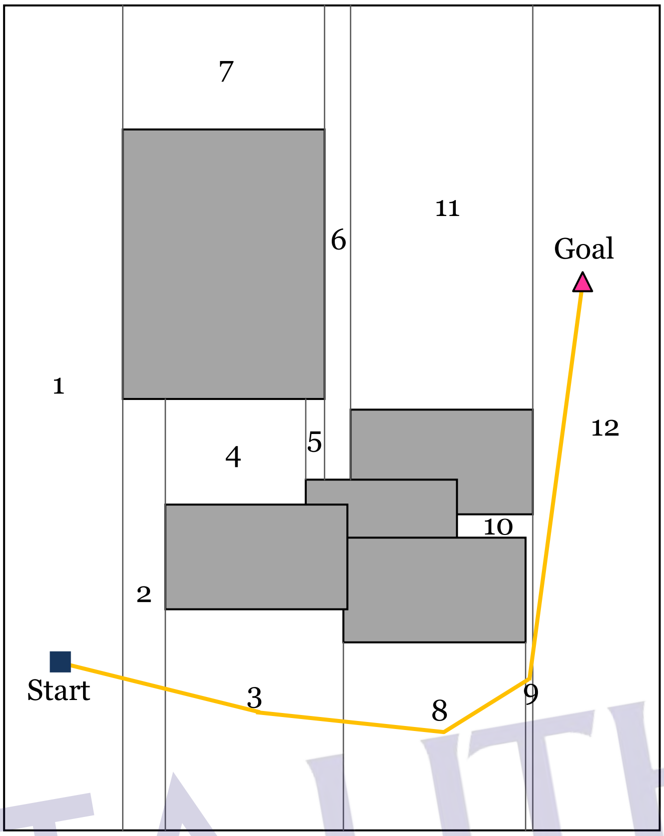


Figure 2.8: Path planning using exact cell decomposition

2.5.5 Related Works Using Cell Decomposition

To date, many researchers have used cell decomposition-based method to solve path planning problems. Brooks and Lozano-Perez [13] developed an algorithm for finding a safe path for a polygonal moving object through cluttered W based on ECD. The obstacles in W are polygonal in shape. The proposed algorithm starts with dividing the C-space into rectangloids with sides normal to the axes of the workspace. The rectangloid are then labelled based on the presence of obstacles in them. The labelled are either empty, full or mixed. Next a path is found using a graph search algorithm through empty rectangloids. If no path is found, mixed rectangloids would be considered to find a path.

Parsons and Canny [26] used cell decomposition-based algorithm for path planning of multiple mobile robots, which share the same workspace. The algorithm computes a path for each robot and capable of avoiding any obstacle and other robots. The cell decomposition algorithm is based on the idea of a product operation that is defined on the cells in a decomposition of a 2D free space. However, the developed algorithm is only useful when infrequent changes occur in obstacles set.

In order to overcome the jagged path using CD, Chen *et al*. [19] introduced framed-quad-tree to create a map. Additionally, the proposed addresses a problem of finding a conditional shortest path through an unknown environment in real time. Conditional shortest path is the path that has shortest path among all possible paths based on known environmental information. The path is found using a propagated circular path planning wave based on a graph search algorithm. Szczerba *et al.* [25] extended the work of [19] using the same approach. In addition to finding a shortest path, they minimised the number of path's turns in 2D environment.

Jun and D'Andrea [27] used approximate cell decomposition-based method to accomplish a robot path planning task. The proposed approach uses initial information of the locations and shapes of the obstacles. The proposed method decomposes the region into uniform cells, and changes the values of probabilities when detecting unexpected changes during the mission. A search algorithm is used to find the shortest path. The drawback of this method is if the penalty is considered for accelerations and decelerations, the graph will become a tree and is expanded exponentially with the number of cells making them very slow. In order to address this issue, Lingelbach [28] applied the so-called Probabilistic Cell Decomposition (PCD) method for path planning in a high-dimensional static C-space as it is easily scalable. Experimental results showed that PCD performed well under various conditions and produced fairly good results for path planning of rigid body movement, maze-like problems and chain-like robotic platform. However, the performance of PCD drops if Q_{free} is significantly smaller than the area of C-space.

Zhang *et.al* [29] utilised ACD for path planning of robots. ACD is used to subdivide the C-space into cells and localised roadmaps are then computed by generating samples within these cells. However, since the ACD's complexity is increased with the number of degree of freedom (DOF) of robots, it is not practical for higher DOF robot. On the other hand, Arney [30] implemented ACD path planning approach, in which the efficiency is achieved by employing a technique found in Geographic Information Systems (GIS) known as tesseral addressing. Each cell is labelled with an address during the decomposition process that defined the cell size, position and neighbors addresses. The planner has *a priori* information about environment and the generated path has an optimal distance from the robot's current location to the target point. It was claimed to be suitable for real time path planning applications. A recent path planning work that is based on cell decomposition-based method is undertaken by Glavaski *et al.* [31]. The objective of the work was to find a path with shortest length through polygonal obstacles using exact cell decomposition.

2.5.6 Roadmap

Roadmap (RM) represents the C-space of obstacles and vehicle with edges and nodes by constructing graphs or maps. A graph G is made of a set of vertices or nodes $V(G)$ as well as a set of edges/lines $E(G)$. $E(G)$ is an unordered pair of distinct vertices of G [32]. RM typically takes several steps to build a graph or map for path planning purpose, starting with establishment of nodes' connections with edges within the free C-space area. Next the starting point p_{start} and target point p_{target} of the vehicle are combined to the network to complete the graph or map. A collision-free path through a series of line segments is then searched from the p_{start} to p_{target} [33] using graph search algorithm. Roadmaps overweigh the cell decompositions method in the number of nodes as path planner needs to search through (in cell decomposition method) in order to find a path. Various methods under the roadmaps methods are discussed in the following sections.

2.5.7 Voronoi Diagram

Voronoi diagram (VD) is a popular roadmap method in path planning. The idea behind VD is to generate a line segment called Voronoi Edge (Ed) which is equidistant from all the points of the obstacle area QO in C-space as shown in Figure 2.8. The point where Ed joins is called Voronoi Vertex (Vd). An example of VD representation, used for path planning is shown in Figure. 2.9. The resultant path is shown in solid black line. As illustrated in the figure, VD has edges that give a maximum clearance path among set of obstacles in the C-space. If a vehicle traverses the planned path, it is guaranteed that the vehicle will not intersect any obstacle. One major disadvantage of VD is the generated paths are not optimal in terms of path length as it produces path which is undesirably long at areas where obstacles are far apart. In addition, the path are also has many unnecessary turns.

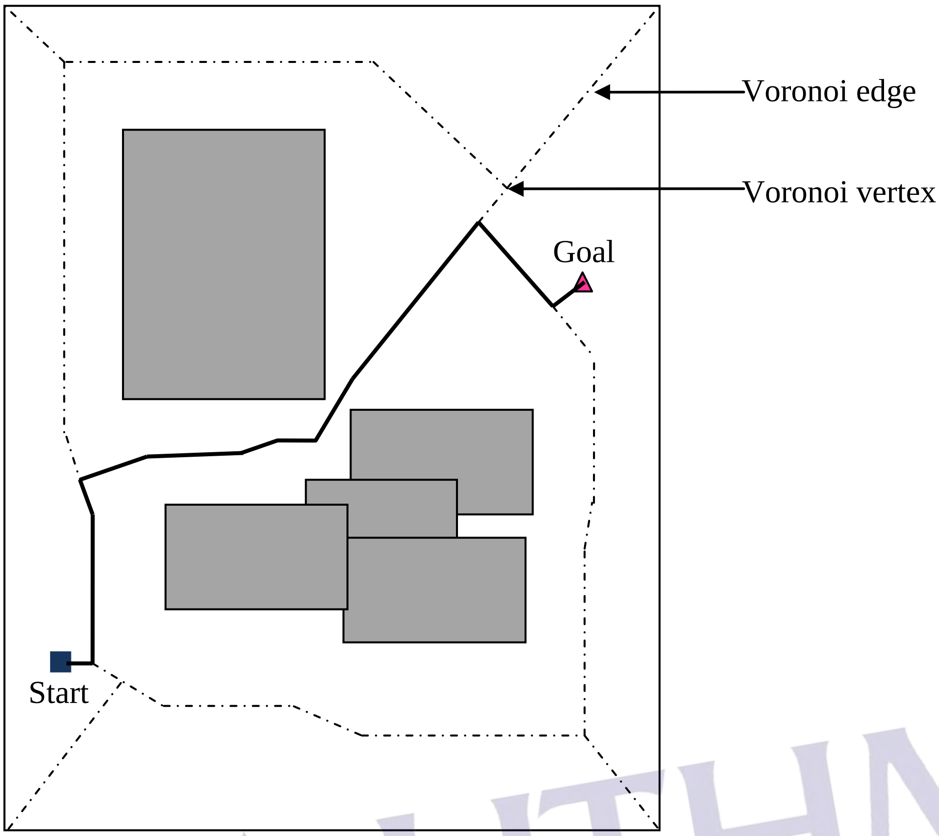


Figure 2.9: A Voronoi diagram. The dashed lines are the set of points equidistant to obstacles. The path is shown in solid darker lines.

2.5.8 Related Works Using Voronoi Diagram

Bortoff [33] proposed a two step path-planning algorithm for UAVs using Voronoi diagram (VD). The first step is to generate a sub-optimal path from VD. In the second step, based on the sub-optimal path, a set of nonlinear ordinary differential equations (ODEs) are simulated. ODEs are used to illustrate the dynamics of virtual masses situated along the planned path. As a result, the generated virtual forces lead the masses away from the nodes and toward each other. The approach could minimizes the path length, and suitable for real-time implementation in dynamic environments with pop-up threats.

Garrido *et al.* [34] used a two-step approach based on VD to find a global shortest path. The diagram is constructed from the environment captured by sensor. The first step extracts the safest area in the environment using VD. In the second step, a shortest path is found using a method called Fast Marching. The description of Fast Marching method can be found in [34].

Xiao *et al.* [19] introduced the local path re-planning algorithm based on the improved VD for an Unmanned Combat Aerial Vehicle (UCAV) with constant speed through a set of threats. The threats in the environment are known beforehand and they are treated as points. The algorithm consists of two steps. The first step is to divide the threats into group where each group contains three adjacent threats. In the second step, a set of waypoints is determined from each group of threats. Each threat is given a unique grade, which in turn produces a path that is not equidistant between two threats. In order to make it suitable for real-time application, only local area of VD is reconstructed every time a new threat appears while the other part of the diagram is unaffected.

Wein *et al*. [35] created a novel diagram known as Visibility-Voronoi diagram for clearance c (denoted by $VV^{(c)}$) to generate an optimal path from a starting to target points with a specific clearance value. Clearance is defined as the distance between the path and obstacles edges. $VV^{(c)}$ is a combination of visibility graph (VG) and VD. As c value increases from 0 to ∞, the resulted graph evolves from VG to the VD. Note that $VV^{(0)}$ is VG while $VV^{(\infty)}$ is VD. In addition, an algorithm to pre-process a scene of QO and construct a data structure called the *VV*-complex was also proposed by [35]. The purpose of *VV*-complex is to efficiently plan paths using any starting and target points as well as any c value. Although the path is smooth and optimal, the technique however does not consider kinematic constraints in planning the path.

Bhattacharya and Gavrilova [36] developed a Voronoi diagram-based algorithm to compute an optimal path from a starting point to a target point in an environment with polygonal obstacles. The algorithms first construct VD of the obstacles and any Voronoi edges that have clearances less than the minimum clearance. In this technique, the clearance can be adjusted to minimize the total path length. Then the algorithm finds a path using a search algorithm. The resultant path is then refined by deleting unnecessary turns. Several more steps have to be executed to make the path optimal. Although the final path is optimal, the technique is too complicated to apply.

A recent study on VD-based path planning is undertaken by Zhang and Wang [37] who worked on mobile robot in an environment with cluttered obstacles. Three different search algorithms are combined to find a globally optimal path in the environment. The proposed approach is faster in obstacle-rich environments in comparison with other search algorithm techniques. Another recent work on path planning for mobile robot applying VD approach was performed by Shao and Lee [30]. They used VD to find a smooth path, then applied pure pursuit path tracking method to satisfy the robot kinematic constraints.

2.5.9 Probabilistic Roadmap

Probabilistic Roadmap (PRM) is a popular method for path planning as it is easy to apply. It makes path planning in large or high-dimensional spaces tractable and provides a good approximation of the connectivity of the configuration space area $\mathcal{Q}_{\text{free}}$. This method consists of two phases i.e. learning phase and query phase. Learning phase constructs a graph G whose nodes are on the free $\mathcal{Q}_{\text{free}}$ and the edges connect the nodes without intersecting any obstacle.

On the other hand, query phase connects the starting point p_{start} and target point p_{target} to G. A search algorithm is then used to find a path from p_{start} to p_{target}. Fig. 2.10 shows an example of PRM used in path planning. A path connecting the starting point and target point is illustrated in solid black line. However, the construction of roadmap is computationally expensive as it might sample thousands of nodes to ensure a path exists. Furthermore the generated path often has poor quality as a result of the randomness inherent in the graph that represents the Q_{free} connectivity. This method may also be incomplete i.e. do not find a path between two locations although there exist a path connecting them, in the present of narrow passage. In addition, there is no way to know whether a path exists as long as no path has been found [38].

REFERENCES

[1]. T. Schouwenaars, B. DeMoor, E. Feron and J. How, "Mixed integer programming for safe multi-vehicle cooperative path planning," ECC, 2001.

[2]. R.Yue and J. Xiao, "Modeling and Path Planning of the City-Climber robot Part II: 3D Path Planning Using Mixed Integer Linear Programming" in International Conference on Robotics and Biomimetic, 2009 © 978-1-4244-4775-6/09/$25.00

[3]. M.G. Earl and R. D'Andrea, "Iterative MILP Methods for Vehicle Control Problems," unpublished.

[4]. Y. Chen, J. Han and L. Yang, "Path Planning of Mobile Robot in Relative Velocity Coordinates," unplubished.

[5]. A. Chaudhry, K. Misovec and R. D'Andrea, "Low Observability Path Planning for an Unmanned Air Vehicle Using Mixed Integer Linear Programming," in 43rd IEEE Conference on Decision and Control, 2004 © 0-7803-8682-5/04/$20.00.

[6]. A. Howard, M.J MatariC and G.S.Sukhatme, An Incremental Self Deployment Algorithm for mobile Sensor Networks, autnomus robots, special Issue on Intelligent Embedded Systems, 13(2), September 2002, pp. 113-126.

[7]. Wang, Y.; Lane, D. M. & Falconer, G. J. (2000). Two novel approaches for unmanned underwater vehicle path planning: constrained optimisation

and semi-infinite constrained optimization. *Robotica*, Vol.18, No.2, pp.123-142, ISSN 02635747

[8]. Zu, D.; Han, J. & Tan, D. (2006). Acceleration Space LP for the Path Planning of Dynamic Target Pursuit and Obstacle Avoidance. *Proceedings of the 6th World Congress on Intelligent Control and Automation*, Vol.2, pp.9084-9088, ISBN-10 1424403324, Dalian, China, June 21-23, 2006

[9]. H. P. Williams and S. C. Brailsford, "Computational Logic and Integer Programming," in Advances in Linear and Integer Programming, Editor J. E. Beasley, Clarendon Press, Oxford, 1996, pp. 249–281.

[10]. A. Bemporad and M. Morari, "Control of Systems Integrating Logic, Dynamics, and Constraints," in Automatica, Pergamon / Elsevier Science, New York NY, Vol. 35, 1999, pp. 407–427.

[11]. C. A. Floudas Nonlinear and Mixed-Integer Programming – Fundamentals and Applications, Oxford University Press, New York NY, 1995 pp. 95–107.

[12]. R. Subramanian, R. Scheff, Jr., J. Quillinan, D. Wiper, and R. Marsten, "Cold start: Fleet assignment at Delta Air Lines," Interfaces, Institute for Operations Research and Management Sciences, Linthicum MD, 24(1), Jan.-Feb. 1994, pp. 104–20.

[13]. ILOG AMPL CPLEX System Version 7.0 User's Guide, ILOG, Incline Village, NV, 2000, pp. 17–53,

[14]. O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 500-505, 1985.

[15]. J. Giesbrecht and Defence R&D Canada. Path planning for unmanned ground

vehicles. *Technical Memorandum DRDC Suffield TM 2004-272*, 2004.

[16]. H. Choset, G. Kantor, W. Burgard, L. Kavraki and S. Thrun. Principles of robot motion: Theory, algorithms, and implementations, The MIT Press, 2005.

[17]. K. H. Yong and N. Ahuja. A potential field approach to path planning. In *IEEE Transactions on Robotics and Automation*, pages 23-32, 1992.

[18]. A. Poty, M. Melchior and A. Oustaloup. Dynamic path planning for mobile robots using fractional potential field. In *First International Symposium on Control, Communications and Signal Processing*, pages 557-561, 2004.

[19]. D. Chen, L. Zhan, X. Chen, Mobile robot path planning based on behaviour information potential field in unknown environments. In *Proceedings of the IEEE International Conference on Robotics and Biomimetic*, pages 683–687, 2004.

[20]. P. Khosla and R. Volpe. Super quadratic artificial potentials for obstacle avoidance and approach. In *Proceedings of the IEEE Conference on Robotics and Automation*,1988.

[21]. J. O. Kim and P. K. Khosla. Real-time obstacles avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation*, 8(3), pages 338-349, 1992.

[22]. S. Akishita, T. Hisanobu and S. Kawamura. Fast path planning available for moving obstacle avoidance by use of laplace potential. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 673-678, 1993.

[23]. R. Daily and D. M. Bevly. Harmonic potential field path planning for high speed vehicles. In *American Control Conference,* pages, 4609-4614, 2008.

[24]. S. J. Russell and P. Norvig. Artificial intelligence: A modern approach, 2nd Edition, Prentice Hall, Upper Saddle River, N.J., 2003.

[25]. R. J Szczerba, D. Z. Chen and K. S. Klenk. Minimum turns/shortest path problems: A framed-subspace approach. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pages 398-403, 1997.

[26]. D. Parsons and J. Canny. A motion planner for multiple mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 8-13, 1990.

[27]. M. Jun and R. D'Andrea. Path planning for unmanned aerial vehicles in uncertain and adversarial environments, cooperative control: Models, applications and algorithms, Kluwer, 2002.

[28]. F. Lingelbach. Path planning using probabilistic cell decomposition. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 467-472, 2004.

[29]. L. Zhang, J. Y. Kim and D. Manocha. A hybrid approach for complete motion planning. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 7-14, 2007.

[30]. T. Arney. An efficient solution to autonomous path planning by approximate cell decomposition. In *Proceedings of the IEEE International Conference on Information and Automation for Sustainability*, pages 88-93, 2007.

[31]. D. Glavaski, M. Volf and M. Bonkovic. Robot motion planning using exact cell decomposition and potential field methods. In *Proceedings of the WSEAS International Conference on Simulation, Modelling and Optimization*, pages 126-131, 2009.

[32]. C. Godsil and G. Royle. Algebraic Graph Theory, Springer, 2001

[33]. R. Siegwart and I. R. Nourbakhsh. Introduction to autonomous mobile robots, Bradford Company, Scituate, MA, USA, 2004.

[34]. S. A. Bortoff. Path planning for UAVs. In *Proceedings of the American Control Conference Chicago Illinois*, pages 364-368, 2000.

[35]. S. Garrido, L. Moreno, M. Abderrahim and F. Martin. Path planning for mobile robot navigation using voronoi diagram and fast marching. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2376-2381, 2006.

[36]. R. Wein, J. P. Van and D. Halperin. The visibility-voronoi complex and its application. In *Computational Geometry: Theory Applications* 36(1), pages 66-87, 2007.

[37]. P. Bhattacharya and M. L. Gavilova. Voronoi diagram in optimal path planning. In *Proceedings of the IEEE International Symposium on Voronoi Diagrams in Science and Engineering*, pages 38-47, 2007.

[38]. Q. Zhang and X. Wang. Global path planning method in uncertain environment. In *Proceedings of the IEEE International Conference on Control Applications*, pages 2725-2730, 2006.

[39]. J. L. Latombe. Motion planning: A journey of robots, molecules, digital actors, and other artefacts. In *International Journal of Robotics Research*, 18(11), pages 1119-1128, 1999.

[40]. L. E Kavraki, P. Svestka, J-C. Latombe and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. In *IEEE Transactions on Robotics and Automation*, 12(4) pages 566-580, 1996.

[41]. G. Song, S. Thomas and N. M. Amato. A general framework for PRM motion planning. In *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 4445-4450, 2003.

[42]. K. Belghith, F. Kabanza, Hartman and R. Nkambou. Anytime dynamic path-planning with flexible probabilistic roadmaps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2372-2377,

2006.

[43]. E. Plaku, K. E. Bekris, B. Y. Chen, A. M. Ladd, and L. E. Kavraki. Sampling-based roadmap of trees for parallel motion planning, In *IEEE Transactions on Robotics*, 21(4), pages 597-608, 2005.

[44]. T. Siméon, J-P. Laumond and C. Nissoux. Visibility-based probabilistic roadmaps for motion planning. In *Advanced Robotics* 14(6). Pages 477-494, 2000.

[45]. Y. Tian, L. Yan, G. Y. Park, S. H. Yang, Y. S. Kim and S. R. Lee, C-Y. Lee. Application of RRT-based local path planning algorithm in unknown environment. In *International Symposium on Computational Intelligence in Robotics and Automation*, pages 456-460, 2007.

[46]. J. Kim and J. P. Ostrowski. Motion planning of aerial robot using rapidly-exploring random trees with dynamic constraints. In *Proceedings of the IEEE International Conference on Robotics &Automation*, pages 2200-2205, 2003.

[47]. N. A. Melchior and R. Simmons. Particle RRT for path planning with uncertainty. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1617-1624. 2007.

[48]. M. Kothari, I. Postlethwaite and D.-W. Gu. Multi-UAV path planning in obstacle rich environments using rapidly-exploring random trees. In *Proceedings of the IEEE Conference on Decision and Control, held jointly with the 28th Chinese Control Conference*, pages 3069-3074, 2009.